Algorithm 959: VBF: A Library of C++ Classes for Vector Boolean Functions in Cryptography

JOSÉ ANTONIO ÁLVAREZ-CUBERO and PEDRO J. ZUFIRIA

VBF is a collection of C++ classes designed for analyzing vector Boolean functions (functions that map a Boolean vector to another Boolean vector) from a cryptographic perspective. This implementation uses the NTL library from Victor Shoup, adding new modules that call NTL functions and complement the existing ones, making it better suited to cryptography. The class representing a vector Boolean function can be initialized by several alternative types of data structures such as *Truth Table*, *Trace Representation*, *and Algebraic Normal Form (ANF)*, among others. The most relevant cryptographic criteria for both block and stream ciphers as well as for hash functions can be evaluated with VBF: it obtains the nonlinearity, linearity distance, algebraic degree, linear structures, and frequency distribution of the absolute values of the Walsh Spectrum or the Autocorrelation Spectrum, among others. In addition, operations such as equality testing, composition, inversion, sum, direct sum, bricklayering (parallel application of vector Boolean functions as employed in Rijndael cipher), and adding coordinate functions of two vector Boolean functions are presented. Finally, three real applications of the library are described: the first one analyzes the KASUMI block cipher, the second one analyzes the Mini-AES cipher, and the third one finds Boolean functions with very high nonlinearity, a key property for robustness against linear attacks.

1. INTRODUCTION

Nowadays, vector Boolean functions (functions that map a Boolean vector to another Boolean vector) play an important role in various fields of science and engineering, such as Coding Theory [MacWilliams and Sloane 1977], Switching Theory [Davio et al. 1978], and Cryptography [Carlet 2008a, 2008c]. Conventional secret key cryptosystems can be expressed as a certain composition of vector Boolean functions. Thus, in cipher design, it is essential to define criteria that measure the cryptographic strength (i.e.,

the robustness against attacks) of Boolean and vector Boolean functions. Moreover, because of the size and complexity of modern ciphers, automatic analysis programs are very helpful in reducing the time required to study cryptographic properties of vector Boolean functions.

In this article, a library of C++ classes for analyzing cryptographic properties of vector Boolean functions (VBFs) is presented. This library is called VBF and uses some modules from the well-known Number Theory Library (NTL) implemented by Victor Shoup (VBF works with any version of NTL, up to the latest one [NTL 2015]). A preliminary version of VBF, lacking several of the modules and features in the current package, was presented in Álvarez-Cubero and Zufiria [2010]. NTL is a high-performance, portable C++ library providing data structures and algorithms for manipulating signed, arbitrary length integers, as well as vectors, matrices, and polynomials over the integers and over finite fields. The decision to use this library is mainly based on four reasons:

(1) It is free software, and may be used according to the terms of the GNU General Public License.
(2) It provides high-quality implementations of state-of-the-art algorithms for the Galois field of order 2.
(3) It may be easily installed in a wide range of platforms.
(4) It provides a clean and consistent interface to a large variety of classes representing mathematical objects that are useful in cryptology.

The VBF library makes use of all the Boolean mathematical objects defined in NTL modules as a starting point. However, it necessarily introduces several new algorithms and structures associated with cryptographic criteria in order to address the characterization of real systems, as shown later in this article.

The main advantages of this approach are derived from the object-oriented implementation and the use of effective algorithms; such advantages are reusability, maintainability, extensibility, and flexibility in the analysis of a broad range of vector Boolean functions employed in symmetric ciphers. The size of the vector Boolean functions that can be analyzed by VBF is restricted by the computational resources (memory, disk space, CPU, etc.) of the platform on which it is executed. However, the maximum value for n and m to be handled by the different functions is conditioned by the maximum value attainable by long int variables (for the computer employed in this work, it is approximately 2^{30}, so that $n_{\max} = m_{\max} \approx 30$). Note that, although these size functions would be compatible with the VBF resource management procedures, the runtime requirements for computing the characteristics would exceed any realistic bound. In order to illustrate VBF applicability in this article, S-boxes used in modern symmetric primitives are studied: a modern cipher used very widely in mobile communications called KASUMI [3rd Generation Partnership Project 2005] is characterized together with a Mini version of the AES (Advanced Encryption Standard); also, highly nonlinear Boolean functions robust against linear attacks are designed.

At the present time, several other packages are available for analyzing vector Boolean functions from the cryptographical point of view, for example:

(1) CRYPTOOL [2014] is a free, open-source e-learning application, used in the implementation and analysis of cryptographic algorithms. It provides cryptanalytical measurement methods (entropy, n-grams, autocorrelation, etc.), but it does not allow the calculation of cryptographic criteria. The current release version, CrypTool 2, is based on the latest .NET Framework (currently .NET 4.0) and it has a pure-plugin architecture. There is also another project called JCrypTool developed in Java and based on Eclipse RCP.

(2) MatPack [2006] is a C++ numerics and graphics library implementing computational methods that are needed in engineering. The cryptographic algorithms are included in the commercial library; these can only be used to analyze some cryptographic properties of Boolean functions and do not address vector Boolean functions.
(3) In Bibliowicz et al. [2003], a system for assisting in the analysis of some criteria of DES-like ciphers is described. This system analyzes only a small subset of the criteria considered by VBF.
(4) bma [Pommerening 2005a, 2005b] outputs the value table, Walsh Spectrum (WS) (a generalized Fourier spectrum), linear profile, differential profile, and some linearity/nonlinearity measures, given the ANF of a vector Boolean function. It is an open-source executable program written in C, computationally very efficient for specific S-boxes analysis.
(5) The boolfun package [BOOLFUN 2010] is open-source software, written in R, to assess cryptographic properties of Boolean functions. It implements three representations: Truth Table, ANF, and WS. It can calculate cryptographic properties of Boolean functions that are relevant for the design of stream ciphers (i.e., cryptographic pseudo-random generators), namely, nonlinearity, algebraic immunity, correlation immunity, and resiliency. Unfortunately, it does not provide specific tools for analyzing vector functions.
(6) SAGE [2014] is free open-source mathematical software that supports research and teaching in algebra, geometry, number theory, cryptography, and related areas. The Cryptography module contains some descriptions of classical ciphers and simplified modern ciphers such as Simplified DES and Mini-AES. Compared with the VBF library, SAGE lacks much useful functionality.

In summary, the packages cited present one (or more) of the following disadvantages: they are commercial, they do not benefit from the new paradigms of object orientation and generic programming, and/or they do not cover the broad spectrum of representation and cryptographic criteria for both Boolean and vector Boolean functions that VBF does. The aim of the VBF package presented in this article is to provide an easy-to-use tool both for the designer and the cryptanalyst of symmetric ciphers. The user only needs to code the basic features related to the vector Boolean functions associated with a cipher (e.g., Truth Table, ANF table, polynomial in ANF, etc.).

This article is structured as follows: Section 2 is devoted to the presentation of the main vector Boolean function concepts. Section 3 starts with the general principles that influenced the design of VBF and describes the VBF classes related to the initialization, cryptographic criteria, and operations over vector Boolean functions. Section 4 gives examples of applications of the VBF framework, analyzing the modern cipher KASUMI together with the Mini-AES, and designing highly nonlinear Boolean functions. Finally, concluding remarks are summarized in Section 5.

2. PRELIMINARIES

The mathematical theory of vector Boolean functions starts with the formal definition of vector spaces whose elements (vectors) have binary elements. Let $< \mathrm{GF}(2), +, \cdot >$ be the finite field of order 2, where $\mathrm{GF}(2) = \mathbb{Z}_2 = \{0, 1\}$, "+" the "integer addition modulo 2," and "$\cdot$" the "integer multiplication modulo 2." $\mathrm{V_n}$ is the vector space of n-tuples of elements from GF(2). The *direct sum* of $\mathbf{x} \in \mathrm{V_{n_1}}$ and $\mathbf{y} \in \mathrm{V_{n_2}}$ is defined as $\mathbf{x} \oplus \mathbf{y} = (x_1, \ldots, x_{n_1}, y_1, \ldots, y_{n_2}) \in \mathrm{V_{n_1+n_2}}$. The *inner product* of $\mathbf{x}, \mathbf{y} \in \mathrm{V_n}$ is denoted by $\mathbf{x} \cdot \mathbf{y}$, and the inner product of real vectors $\mathbf{x}, \mathbf{y} \in \mathbb{R}^n$ is denoted by $\langle \mathbf{x}, \mathbf{y} \rangle$.

One can now define binary functions between this type of vector spaces, whose linearity analysis (for robustness-against-attacks purposes) becomes very important.

$f : \mathrm{V_n} \to \mathrm{GF}(2)$ is called a *Boolean function* and $\mathcal{F}_n$ is the set of all Boolean functions on $\mathrm{V_n}$. $\mathcal{L}_n$ is the set of all linear Boolean functions on $\mathrm{V_n}$: $\mathcal{L}_n = \{l_\mathbf{u} \; \forall \mathbf{u} \in \mathrm{V_n} \mid l_\mathbf{u}(\mathbf{x}) = \mathbf{u} \cdot \mathbf{x}\}$ and $\mathcal{A}_n$ is the set of all affine Boolean functions on $\mathrm{V_n}$.

It is possible to characterize Boolean functions via alternative and very useful associated mappings. In the following, some of these mappings are presented. The real-valued mapping $\chi_\mathbf{u}(\mathbf{x}) = (-1)^{\sum_{i=1}^{i=n} u_i x_i} = (-1)^{\mathbf{u}\cdot\mathbf{x}}$ for $\mathbf{x}, \mathbf{u} \in \mathrm{V_n}$ is called a *character*. The character form of $f \in \mathcal{F}_n$ is defined as $\chi_f(\mathbf{x}) = (-1)^{f(\mathbf{x})}$. The Truth Table of χ_f is defined as the $(1, -1)$-*sequence vector* or *sequence vector* of f and is denoted by $\xi_f \in \mathbb{R}^{2^n}$.

Let $f \in \mathcal{F}_n$ be a Boolean function; the *Walsh transform* of f at $\mathbf{u} \in \mathrm{V_n}$ is an n-dimensional Discrete Fourier Transform and can be obtained as follows:

$$\hat{\chi}_f(\mathbf{u}) = \langle \xi_f, \xi_{l_\mathbf{u}} \rangle = \sum_{\mathbf{x} \in \mathrm{V_n}} (-1)^{f(\mathbf{x}) + \mathbf{u}\mathbf{x}}. \tag{1}$$

The *autocorrelation* of $f \in \mathcal{F}_n$ with respect to the shift $\mathbf{u} \in \mathrm{V_n}$ is a measure of the statistical dependency among the involved variables (indicating robustness against randomness-based attacks). It is the cross-correlation of f with itself, denoted by $r_f(\mathbf{u}) : \mathrm{V_n} \to \mathbb{Z}$ and defined by

$$r_f(\mathbf{u}) = \sum_{\mathbf{x} \in \mathrm{V_n}} \chi_f(\mathbf{x}) \chi_f(\mathbf{x} + \mathbf{u}) = \sum_{\mathbf{x} \in \mathrm{V_n}} (-1)^{f(\mathbf{x}) + f(\mathbf{u} + \mathbf{x})}. \tag{2}$$

The factor $\frac{1}{2^n}$ is generally omitted (see Carlet [2004]).

The *directional derivative* of $f \in \mathcal{F}_n$ in the direction of $\mathbf{u} \in \mathrm{V_n}$ is defined by

$$\Delta_\mathbf{u} f(\mathbf{x}) = f(\mathbf{x} + \mathbf{u}) + f(\mathbf{x}), \quad \mathbf{x} \in \mathrm{V_n}. \tag{3}$$

We call the linear kernel of f the set of those vectors $\mathbf{u}$ such that $\Delta_\mathbf{u} f$ is a constant function. The linear kernel of any Boolean function is a subspace of $\mathrm{V_n}$. Any element $\mathbf{u}$ of the linear kernel of f is said to be a linear structure of f.

Given $f \in \mathcal{F}_n$, a nonzero function $g \in \mathcal{F}_n$ is called an annihilator of f if $fg = 0$.

We now extend the scope of the study by considering functions between any pair of binary-valued vector spaces. $F : \mathrm{V_n} \to \mathrm{V_m}$, $F(\mathbf{x}) = (f_1(\mathbf{x}), \ldots, f_m(\mathbf{x}))$ is called a *vector Boolean function* and $\mathcal{F}_{n,m}$ is the set of all vector Boolean functions $F : \mathrm{V_n} \to \mathrm{V_m}$. Each $f_i : \mathrm{V_n} \to \mathrm{GF}(2) \; \forall i \in \{1, \ldots, m\}$ is a coordinate function of F. The *indicator function* of $F \in \mathcal{F}_{n,m}$, denoted by $\theta_F : \mathrm{V_n} \times \mathrm{V_m} \to \{0, 1\}$, is defined in Chabaud and Vaudenay [1995] as

$$\theta_F(\mathbf{x}, \mathbf{y}) = \begin{cases} 1 & \text{if } \mathbf{y} = F(\mathbf{x}) \\ 0 & \text{if } \mathbf{y} \neq F(\mathbf{x}). \end{cases} \tag{4}$$

Again, several mappings associated with a vector Boolean function can be defined, in similar terms to the case of binary functions. Hence, the character form of $(\mathbf{u}, \mathbf{v}) \in \mathrm{V_n} \times \mathrm{V_m}$ can be defined as follows: $\chi_{(\mathbf{u},\mathbf{v})}(\mathbf{x}, \mathbf{y}) = (-1)^{\mathbf{u}\cdot\mathbf{x} + \mathbf{v}\cdot\mathbf{y}}$. Similarly, let $F \in \mathcal{F}_{n,m}$ be a vector Boolean function; its *Walsh Transform* is the two-dimensional Walsh transform defined by

$$\hat{\theta}_F(\mathbf{u}, \mathbf{v}) = \sum_{\mathbf{x} \in \mathrm{V_n}} \sum_{\mathbf{y} \in \mathrm{V_m}} \theta_F(\mathbf{x}, \mathbf{y}) \chi_{(\mathbf{u},\mathbf{v})}(\mathbf{x}, \mathbf{y}) = \sum_{\mathbf{x} \in \mathrm{V_n}} (-1)^{\mathbf{u}\mathbf{x} + \mathbf{v}F(\mathbf{x})}. \tag{5}$$

Also, the *autocorrelation* of $F \in \mathcal{F}_{n,m}$ with respect to the shift $(\mathbf{u}, \mathbf{v}) \in \mathrm{V_n} \times \mathrm{V_m}$ is the cross-correlation of F with itself, denoted by $r_F(\mathbf{u}, \mathbf{v}) : \mathrm{V_n} \times \mathrm{V_m} \to \mathbb{Z}$, so that [Nyberg 1995]

$$r_F(\mathbf{u}, \mathbf{v}) = \sum_{\mathbf{x} \in \mathrm{V_n}} \chi_{\mathbf{v}F}(\mathbf{x} + \mathbf{u}) \chi_{\mathbf{v}F}(\mathbf{x}) = \sum_{\mathbf{x} \in \mathrm{V_n}} (-1)^{\mathbf{v}F(\mathbf{x}+\mathbf{u}) + \mathbf{v}F(\mathbf{x})}. \tag{6}$$

Let $F \in \mathcal{F}_{n,m}$ and $\mathbf{u} \in \mathrm{V_n}$; then the *difference vector Boolean function* of F in the direction of $\mathbf{u} \in \mathrm{V_n}$, denoted by $\Delta_{\mathbf{u}}F \in \mathcal{F}_{n,m}$, is defined as follows: $\Delta_{\mathbf{u}}F(\mathbf{x}) = F(\mathbf{x}+\mathbf{u}) + F(\mathbf{x})$, $\mathbf{x} \in \mathrm{V_n}$. If the following equality is satisfied: $\Delta_{\mathbf{u}}F(\mathbf{x}) = \mathbf{c}$, $\mathbf{c} \in \mathrm{V_n}$, $\forall \mathbf{x} \in \mathrm{V_n}$, then $\mathbf{u} \in \mathrm{V_n}$ is called a linear structure of F.

Finally, we define the simplifying notation for the maximum of the absolute values of a set of real numbers $\{a_{\mathbf{uv}}\}_{\mathbf{u},\mathbf{v}}$, characterized by vectors $\mathbf{u}$ and $\mathbf{v}$, as $\max(a_{\mathbf{uv}}) = \max_{(\mathbf{u},\mathbf{v})}\{|a_{\mathbf{uv}}|\}$. Using the same simplifying notation, we can define the $\overset{*}{\max}(\cdot)$ operator on a set of real numbers $\{a_{\mathbf{uv}}\}_{\mathbf{u},\mathbf{v}}$, as $\overset{*}{\max}(a_{\mathbf{uv}}) = \max_{(\mathbf{u},\mathbf{v})\neq(\mathbf{0},\mathbf{0})}\{|a_{\mathbf{uv}}|\}$. This notation will be used in some criteria definitions.

3. DESIGN PHILOSOPHY: CLASSES, CRYPTOGRAPHIC CRITERIA, AND OPERATIONS

The core of the VBF library is the VBF class, which represents vector Boolean functions whose data members and member functions make use of the NTL modules listed in Table 1.1 of the User Manual that accompanies the software [VBF 2015]. However, some new cryptography-related member functions were added to the previous modules. New modules, which are not present in NTL, are defined and they are listed in Table 1.2 of VBF [2015].

The main file in the library, called *VBF.h*, has the definitions of the objects described in the next subsection and makes use of the cited modules.

3.1. VBF Class: Initialization and Representations

Here we describe the methods that can be used to represent a VBF class related to a vector Boolean function $F \in \mathcal{F}_{n,m}$. Some of these representations can be used to initialize a VBF class.

(1) *Truth Table*, defined as $\mathsf{T}_F \in \mathsf{M}_{2^n \times m}(\mathrm{GF}(2))$, where

$$\mathsf{T}_F = \begin{bmatrix} f_1(\boldsymbol{\alpha}_0) & \cdots & f_m(\boldsymbol{\alpha}_0) \\ f_1(\boldsymbol{\alpha}_1) & \cdots & f_m(\boldsymbol{\alpha}_1) \\ \cdots & \cdots & \cdots \\ f_1(\boldsymbol{\alpha}_{2^n-1}) & \cdots & f_m(\boldsymbol{\alpha}_{2^n-1}) \end{bmatrix}, \tag{7}$$

satisfying that f_i $i \in \{1, \ldots, m\}$ are its component functions; each $\boldsymbol{\alpha}_i = (x_1, \ldots, x_n) \in \mathrm{V_n}$ $i \in \{1, \ldots, 2^n - 1\}$ is a vector whose decimal equivalent is $dec(\boldsymbol{\alpha}_i) = i = \sum_{j=1}^{n} x_j 2^{n-j}$, and all the vectors of $\mathrm{V_n}$ can be listed so that $\boldsymbol{\alpha}_0 < \boldsymbol{\alpha}_1 < \cdots < \boldsymbol{\alpha}_{2^n-1}$.

The Truth Table for an n-variable Boolean function f should be in lexicographical form, that is, $T_f = (T_f(0), T_f(1), T_f(2), \ldots, T_f(2^{n-1}))$. Since the Truth Table length might be too large, we represent it in hexadecimal rather than in binary notation. The hexadecimal Truth Table is obtained by replacing each 4 bits by their corresponding hexadecimal form. For instance, to enter $f = (0, 0, 1, 1, 1, 1, 1, 1)$, one should just write $3F$.

(2) *Trace representation.* When $m = n$, we endow $\mathrm{V_n}$ with the structure of the field $\mathrm{GF}(2^n)$. Any $F \in \mathcal{F}_{n,n}$ admits a unique *univariate polynomial representation* over $\mathrm{GF}(2^n)$, of degree at most $2^n - 1$:

$$F(\mathbf{x}) = \sum_{i=0}^{2^n-1} \delta_i x^i, \ \delta_i \in \mathrm{GF}(2^n). \tag{8}$$

A general way to derive this polynomial representation is given by a Lagrange interpolation from the knowledge of the irreducible polynomial of degree n over $GF(2)$ associated with the field $GF(2^n)$ and the Truth Table of F.

The *interpolation attack* [Jakobsen and Knudsen 1997] is efficient when the degree of the univariate polynomial representation of the S-box over GF(2^n) is low or when the distance of the S-box to the set of low univariate degree functions is small. This attack exploits the low degree of the algebraic relation between some input (output, respectively) and intermediate data to infer some keybits relating the output (input, respectively) and the intermediate data.

(3) *Polynomials in ANF.* F can be uniquely represented by m multivariate polynomials over GF(2) (called coordinate functions), where each variable has power at most one. Each of these polynomials can be expressed as a sum of all distinct kth-order product terms ($0 < k \leq n$) of the variables in the form

$$f(x_1, \ldots, x_n) = a_0 + a_1 x_1 + \cdots + a_n x_n + a_{12} x_1 x_2 + \cdots + a_{n-1,n} x_{n-1} x_n + \cdots + a_{12\ldots n} x_1 x_2 \ldots x_n = \sum_{\mathsf{I} \in \mathsf{P}(N)} a_{\mathsf{I}} \left(\prod_{i \in \mathsf{I}} x_i \right) = \sum_{\mathsf{I} \in \mathsf{P}(N)} a_{\mathsf{I}} x^{\mathsf{I}}, \quad a_{\mathsf{I}} \in \mathrm{GF}(2), \tag{9}$$

where $\mathsf{P}(N)$ denotes the power set of $N = \{1, \ldots, n\}$. This representation of f is called the *algebraic normal form (ANF)* of f.

(4) *ANF table* of F, denoted by $\mathsf{ANF}_F \in \mathsf{M}_{2^n \times m}(\mathrm{GF}(2))$, represents the 2^n coefficients of the polynomials of each of the m coordinate functions in *ANF*.

(5) *Characteristic function*, which is a matrix whose rows are indexed by $\mathbf{x} \in \mathrm{V_n}$ and whose columns are indexed by $\mathbf{y} \in \mathrm{V_m}$ in lexicographic order, is denoted by $\mathsf{Img(F)} \in \mathsf{M}_{2^n \times 2^m}(\mathrm{GF}(2))$ and defined as follows:

$$\mathsf{Img(F)} = \begin{bmatrix} \theta_F(\boldsymbol{\alpha}_0, \boldsymbol{\alpha}_0) & \cdots & \theta_F(\boldsymbol{\alpha}_0, \boldsymbol{\alpha}_{2^m-1}) \\ \theta_F(\boldsymbol{\alpha}_1, \boldsymbol{\alpha}_0) & \cdots & \theta_F(\boldsymbol{\alpha}_1, \boldsymbol{\alpha}_{2^m-1}) \\ \cdots & \cdots & \cdots \\ \theta_F(\boldsymbol{\alpha}_{2^n-1}, \boldsymbol{\alpha}_0) & \cdots & \theta_F(\boldsymbol{\alpha}_{2^n-1}, \boldsymbol{\alpha}_{2^m-1}) \end{bmatrix}, \tag{10}$$

where $\theta_F(\mathbf{x}, \mathbf{y})$ stands for the indicator function defined in Equation (4).

(6) *Walsh Spectrum*, which is a matrix whose rows are characterized by $\mathbf{u} \in \mathrm{V_n}$ and whose columns are characterized by $\mathbf{v} \in \mathrm{V_m}$ in lexicographic order, is denoted by $\mathsf{WS}(F) \in \mathsf{M}_{2^n \times 2^m}(\mathbb{R})$. It holds that $\hat{\theta}_F(\mathbf{u}, \mathbf{v}) = \mathsf{WS}(F)(\mathbf{u}, \mathbf{v})$.

(7) *Linear profile*, which is a matrix whose rows are characterized by $\mathbf{u} \in \mathrm{V_n}$ and whose columns are characterized by $\mathbf{v} \in \mathrm{V_m}$ in lexicographic order, is denoted by $\mathsf{LP}(F) \in \mathsf{M}_{2^n \times 2^m}(\mathbb{R})$. It holds that $\mathsf{LP}(F)(\mathbf{u}, \mathbf{v}) = \frac{1}{2^{n+m}} |\mathsf{WS}(F)(\mathbf{u}, \mathbf{v})|^2$.

(8) *Differential Profile*, denoted by $\mathsf{DP}(F) \in \mathsf{M}_{2^n \times 2^m}(\mathbb{R})$, results from the application of the Walsh transform to the linear profile.

(9) *Autocorrelation Spectrum*, denoted by $\mathsf{R}(F) \in \mathsf{M}_{2^n \times 2^m}(\mathbb{Z})$, is obtained by Equation (6).

(10) *Permutation vector.* If F is a Boolean permutation, that is, it is bijective and has the same number of input bits as output bits ($n = m$), then it can be defined as a vector: $v = (F(1), \ldots, F(n))$, where $F(i)$ is the image of the bit i for F.

(11) *Expansion and Compression DES permutations.* If F is an affine vector Boolean function with $n \neq m$ in the form of Expansion and Compression DES permutations [National Institute of Standards and Technology 1999], then it can be defined as an array with m elements, which are the output bits.

(12) *DES-like S-box representation.* The VBF class also supports the definition of F as given in National Institute of Standards and Technology [1999] for the DES S-boxes.

3.2. Cryptographic Criteria

The cryptographic criteria that can be evaluated by means of the VBF class are the following:

(1) *Nonlinearity*, defined as the minimum among the nonlinearities of all nonzero linear combinations of the coordinate functions of F; it can be evaluated from the Walsh Spectrum in the following way:

$$\mathcal{NL}(F) = \min_{\mathbf{v}\neq\mathbf{0}\in \mathrm{V_m}} \mathcal{NL}(\mathbf{v}\cdot F) = 2^{n-1} - \frac{1}{2} \overset{*}{\max} (\mathrm{WS}(F)(\mathbf{u}, \mathbf{v})), \tag{11}$$

where the same symbol $\mathcal{NL}$ is employed to denote the nonlinearity of either a scalar or a vector function. This criterion is a measure of the distance of a vector Boolean function and all affine vector Boolean functions. If this distance is small, it is possible to mount affine approximations of the vector Boolean functions involved in a cryptosystem to build attacks on this system [Matsui 1994]. Thus, this property is useful to assess the resistance of a vector Boolean function to linear attacks (including correlation attacks), that is, attacks where the function F is approximated by an affine function.

(2) For every positive integer r, the *rth-order nonlinearity* of a vector Boolean function F is the minimum rth-order nonlinearity of its component functions; the rth-order nonlinearity of a Boolean function equals its minimum Hamming distance to functions of algebraic degrees at most r (see Carlet [2008b] for details):

$$\mathcal{NL}_r(F) = \min_{\mathbf{v}\neq\mathbf{0}\in \mathrm{V_m}} \mathcal{NL}_r(\mathbf{v}\cdot F) = \min_{\mathbf{v}\neq\mathbf{0}\in \mathrm{V_m}} \min_{f\in\mathcal{F}_n} d(f, \mathbf{v}\cdot F). \tag{12}$$

Computing rth-order nonlinearity is not an easy task for $r \geq 2$. Unlike the first-order nonlinearity, there are no efficient algorithms to compute second-order nonlinearities for $n \geq 11$. VBF library naive exhaustive search is employed for this purpose.

(3) *Linearity distance*, defined as the minimum among the linearity distances of all nonzero linear combinations of the coordinate functions of F, may be computed from the Differential Profile using

$$\mathcal{LD}(F) = \min_{\mathbf{v}\neq\mathbf{0}\in \mathrm{V_m}} \mathcal{LD}(\mathbf{v}\cdot F). \tag{13}$$

The *linearity distance* of a Boolean function is a characteristic defined by the distance to the set of all Boolean functions admitting nonzero linear structures. These include, among others, all the affine functions and all nonbent quadratic functions and are defined as follows [Meier and Staffelbach 1990]:

$$\mathcal{LD}(f) = d(f, \mathcal{LS}_n) = \min_{\mathrm{S}\in\mathcal{LS}_n} d(f, \mathrm{S}). \tag{14}$$

S-boxes used in block ciphers should have no nonzero linear structures (see Evertse [1988]). The existence of nonzero linear structures, for the functions implemented in stream ciphers, is a potential risk that should also be avoided, despite the fact that such existence could not be used in attacks, so far.

(4) *Balancedness.* $F \in \mathcal{F}_{n,m}$ is balanced (or has balanced output) if each possible output m-tuple occurs with equal probability 2^{-m}; that is, its output is uniformly distributed in $\mathrm{V_m}$. This criterion can be evaluated from the Walsh Spectrum in the following way:

$$\hat{\theta}_F(\mathbf{0}, \mathbf{v}) = 0, \ \forall \mathbf{v} \neq \mathbf{0} \in \mathrm{V_m}. \tag{15}$$

Cryptographic functions must be balanced; that is, their output must be uniformly distributed over $\{0, 1\}$ to avoid statistical dependence between the input and the output (which can be used in attacks).

(5) *Correlation immunity.* $F \in \mathcal{F}_{n,m}$ is an (n, m, t)–$\mathcal{CI}$ function if and only if every nonzero linear combination $f(\mathbf{x}) = \sum_{i=1}^{m} v_i f_i(\mathbf{x})$ of coordinate functions of F is an $(n, 1, t)$–$\mathcal{CI}$

function or a $t - \mathcal{CI}$ function, where $\mathbf{x} \in \mathrm{V_n}$, $v_i \in \mathrm{GF}(2)$ $i = 1, \ldots, m$ and not all zeroes. This criterion can be obtained from the Walsh Spectrum as follows:

$$\hat{\theta}_F(\mathbf{u}, \mathbf{v}) = 0, \ \forall \mathbf{u} \in \mathrm{V_n}, \ 1 \leq wt(\mathbf{u}) \leq t, \ \forall \mathbf{v} \neq \mathbf{0} \in \mathrm{V_m}. \tag{16}$$

A function $f \in \mathcal{F}_n$ is $t - \mathcal{CI}$ if its output is statistically independent of any subset of at most t input bits. Correlation immunity is used to assess the resistance to correlation attacks [Siegenthaler 1985]. Note that the statistical measure used to assess independency between input and output bits is (conditional) mutual information. $F \in \mathcal{F}_{n,m}$ is $t - \mathcal{CI}$ if its output distribution does not change when we fix t variables x_i. A $t - \mathcal{CI}$ function, which is also balanced, is called a t-resilient function. This criterion is related to an attack on pseudo-random generators using combining functions, called a correlation attack. If f is not t-resilient, then there exists a correlation between the output of the function and (at most) t bits of its input; if t is small, f is prone to a divide-and-conquer attack due to Siegenthaler [1985] and later improved by several authors with fast correlation attacks.

For the pseudo-random generators, the best-known cryptanalytic technique is the *correlation attack*, which is based on the idea of finding correlation between the outputs and the inputs, that is, finding S-boxes with low resiliency.

(6) *Propagation.* $F \in \mathcal{F}_{n,m}$ satisfies the propagation criterion of degree l ($\mathcal{PC}(l)$) if any nonzero linear combination of the component Boolean functions satisfies the $\mathcal{PC}(l)$. This criterion can be obtained from the Autocorrelation Spectrum in the following way:

$$r_F(\mathbf{u}, \mathbf{v}) = 0, \ \forall \mathbf{u} \in \mathrm{V_n}, \ 1 \leq wt(\mathbf{u}) \leq l, \ \forall \mathbf{v} \neq \mathbf{0} \in \mathrm{V_m}. \tag{17}$$

This criterion is based on the properties of the derivatives of Boolean functions and describes the behavior of a function whenever some input bits are complemented. $F \in \mathcal{F}_{n,m}$ is said to satisfy the propagation characteristics with respect to $\mathbf{u} \in \mathrm{V_n}$ if and only if $F(\mathbf{x}) + F(\mathbf{x} + \mathbf{u})$ is balanced.

(7) *Global avalanche* is defined by two indicators [Zhang and Zheng 1995]. First, the *absolute indicator* of F, denoted by $\mathcal{MAXAC}(F)$, defines the maximum absolute nonzero value of the Autocorrelation Spectrum and quantifies the distance to the set $\mathcal{LS}_n$. Second, the *sum-of-squares indicator*, denoted by σ, is the second moment of the autocorrelation coefficients. In order to achieve good diffusion, cryptographic functions should achieve low values of both indicators.

(8) *Algebraic degree* is defined as the minimum among the algebraic degrees of all nonzero linear combinations of the coordinate functions of F [Nyberg 1993], namely:

$$\mathcal{DEG}(F) = \min_g \left\{ \mathcal{DEG}(g) \mid g = \sum_{j=1}^{m} v_j f_j, \ \mathbf{v} \neq \mathbf{0} \in \mathrm{V_m} \right\}, \tag{18}$$

where the algebraic order or degree of a Boolean function is the order of the largest product term in the *ANF*. This criterion is obtained by generating the ANF table and then analyzing the order of all the linear combinations of coordinate functions. Cryptographic functions must have high algebraic degrees. Indeed, cryptosystems using vector Boolean functions for confusion (S-boxes in block ciphers, combining functions in stream ciphers, etc.) can be attacked if the functions have low degrees. *Higher-order differential attack* [Lai 1994] exploits the fact that the algebraic degree of the S-box is low.

(9) *Algebraic immunity* of a Boolean function $f \in \mathcal{F}_n$ is defined as the minimum degree of all annihilators of f or $1 + f$ and it is denoted by $\mathcal{AI}(f)$ [Courtois 2003; Courtois and Meier 2002; Faugère and Ars 2003]. A function f should not be used if f or

$1 + f$ has a low-degree annihilator. If this happens, algebraic attacks [Courtois and Pieprzyk 2002] can be executed. Algebraic attacks recover the secret key, or at least the initialization of the system, by solving a system of multivariate algebraic equations.

The component algebraic immunity of any $F \in \mathcal{F}_{n,m}$, denoted by $\mathcal{AI}(F)$, is the minimal algebraic immunity of the component functions $\mathbf{v} \cdot F(\mathbf{v} \neq \mathbf{0} \in \mathrm{V_m})$ of the vector Boolean function.

Other useful information in cryptanalysis can be obtained by means of the VBF class (see the member functions in Table 4.2 of VBF [2015]):

(1) The *linear potential* of F, defined as $\mathcal{LP}(F) = \frac{1}{2^{2n}} \cdot \overset{*}{\max}(\mathsf{WS}(F)(\mathbf{u}, \mathbf{v})^2)$, is a measure of linearity in linear cryptanalysis and satisfies [Chabaud and Vaudenay 1995] $2^{-n} \leq \mathcal{LP}(F) \leq 1$ so that the lower bound holds if and only if F has maximum nonlinearity (F is bent) and the upper bound is reached when F is linear or affine. *Linear cryptanalysis* is based on the idea of finding high-probable linear or affine relations between the inputs and outputs of S-boxes present in the cipher, that is, finding S-boxes with low nonlinearity. The attack is mounted by finding high-probability parity of the sum of some input, output, and key bits, and hence deducing one bit of information about the key.

(2) Linear relations associated with a specific value of the linear profile.

(3) The *Differential Potential* of F, defined as $\mathcal{DP}(F) = \overset{*}{\max}(\mathsf{DP}(F)(\mathbf{u}, \mathbf{v}))$, is a measure of the robustness against differential cryptanalysis where $2^{-m} \leq \mathcal{DP}(F) \leq 1$ and the lower bound holds if and only if F is bent and the upper bound is reached when F is linear or affine. The differential uniformity of $F \in \mathcal{F}_{n,m}$ and its differential potential are related by $\mathcal{DP}(F) = 2^{-n}\mathcal{DU}(F)$.

Differential cryptanalysis is based on the idea of finding high-probable differential pairs between the inputs and outputs of S-boxes present in the cipher, that is, finding S-boxes with low linearity distance. Differential cryptanalysis [Biham and Shamir 1990] can be seen as an extension of the ideas of attacks based on the presence of linear structures [Nyberg 1991]. If $\mathbf{u}$ is a linear structure of f, then the inputs of difference $\mathbf{u}$ result in output differences of 1 or -1 with probability 1. In differential cryptanalysis, it is only required that inputs of difference $\Delta\mathbf{x}$ lead to a known difference $\Delta\mathbf{y}$ with high probability, or with a probability that noticeably exceeds the mean, and hence deducing some information about the key.

(4) Differential relations associated with a specific value of the Differential Profile.

(5) The *linear structures* of F, defined as the vectors for which associated rows in the Differential Profile coincide with the vector zero.

Lai [1990] showed that if $f \in \mathcal{F}_n$ has $k < n$ linearly independent vectors $\mathbf{b_1}, \ldots, \mathbf{b_k}$ that are linear structures, then f can be mapped to $g \in \mathcal{F}_n$ via a linear transformation where

$$g(x_1, \ldots, x_n) = m_1 x_1 + \cdots + m_k x_k + g'(x_{k+1}, \ldots, x_n). \tag{19}$$

The cryptanalyst may be able to take advantage of the linear structures in f if some of the m_i, $i = 1, \ldots, k$ in Equation (19) are zero, thus eliminating the influence of some variables (possibly key bits) on the ciphertext.

The cryptanalytic value of linear structures lies in their potential to map a nonlinear function to a degenerate function via a linear transformation, which may reduce the size of the keyspace.

(6) The *maximum possible (for n even) nonlinearity* for a vector Boolean function with the same dimensions as $F \in \mathcal{F}_{n,m}$ (when n is even). The functions with maximum possible nonlinearity are called bent functions and their nonlinearity is equal to

$2^{n-1} - 2^{n/2-1}$ [Rothaus 1976]. The nonlinearity of a vector Boolean function is defined as the minimum among the nonlinearities of all nonzero linear combinations of its coordinate functions. Thus, the maximum possible nonlinearity for a vector Boolean function coincides with the maximum possible nonlinearity for a Boolean function.

(7) The *type of function in terms of nonlinearity* of a Boolean function (only for $m = 1$), that is, if it is linear, almost optimal, or a bent function (F with maximum nonlinearity and n even).

Let f be a Boolean function with n variables. Then f is said to be almost optimal if $\overset{*}{\max}\ (\mathsf{WS}(F)(\mathbf{u}, \mathbf{v})) \leq 2^{(n+1)/2}$ when n is odd, and $\overset{*}{\max}\ (\mathsf{WS}(F)(\mathbf{u}, \mathbf{v})) \leq 2^{(n+1)/2}$ when n is even.

(8) The *maximum possible linearity distance* for a vector Boolean function with the same dimensions as F.

(9) The *frequency distribution of the absolute values of the Walsh Spectrum.*

(10) The *frequency distribution of the absolute values of the Autocorrelation Spectrum.*

(11) The *cycle structure* of an invertible vector Boolean function $F \in \mathcal{F}_{n,n}$ (permutation) describes the number of cycles and their length.

A cycle structure with a low number of cycles of high length is considered well suited to be used in cipher design. This fact means that many transpositions are present.

(12) *Fixed points* of F, that is, $\{\mathbf{x} \mid F(\mathbf{x}) = \mathbf{x}\}$.

(13) *Negated fixed points* of F, that is, $\{\mathbf{x} \mid F(\mathbf{x}) = \overline{\mathbf{x}}\}$.

A cryptographic primitive with a high number of fixed and/or negated fixed points is considered to be not well designed, since it lacks the needed randomness.

A list of the member functions related to these criteria may be found in Tables 4.1 and 4.2 of VBF [2015]. For a detailed explanation of the cryptographic criteria and their properties, see Álvarez-Cubero and Zufiria [2012].

3.3. Operations Over Vector Boolean Functions

In this subsection, the operations over vector Boolean functions supported by the VBF class are described. Some of them correspond to secondary constructions, which build (n, m) variable vector Boolean functions from (n', m') variable ones (with $n' \leq n, m' \leq m$). The direct sum has been used to construct resilient and bent Boolean functions [Carlet 2004]. Adding coordinate functions and bricklayering (also called concatenation) are operations used to build modern ciphers such as CAST [Adams and Tavares 1993], DES [National Institute of Standards and Technology 1999] and AES [Daemen and Rijmen 2002]. Additionally, VBF provides operations for identification if two vector Boolean functions are equal, the sum of two vector Boolean functions and the composition of two vector Boolean functions. The definitions of all the supported operations are as follows:

(1) Let $n \geq 1, m \geq 1$, $F, G \in \mathcal{F}_{n,m}$. F and G are *equal* if their Truth Tables are the same.

(2) Let $F \in \mathcal{F}_{n,p}$, $G \in \mathcal{F}_{p,m}$; then the *composition function* is $G \circ F \in \mathcal{F}_{n,m}$.

(3) Let $n \geq 1$, $F \in \mathcal{F}_{n,n}$. F^{-1} is the *functional inverse* of F if the composition of both functions results in the identity function.

(4) Let $n \geq 1, m \geq 1$, $F, G \in \mathcal{F}_{n,m}$. The *sum* of F and G (denoted by $F + G$) is the vector Boolean function whose Truth Table results from the addition of the Truth Tables of F and G: $T_{F+G} = T_F + T_G$. It can be proved that the Walsh Spectrum of the sum can be obtained by the convolution of the columns vectors of the respective Walsh Spectra.

(5) Let $n = n_1 + n_2, n_1, n_2 \geq 1, m \geq 1, F \in \mathcal{F}_{n_1,m}$, and $G \in \mathcal{F}_{n_2,m}$. The *Direct Sum* of F and G is the function

$$\begin{aligned}(F \oplus G) &: \mathrm{V_{n_1}} \times \mathrm{V_{n_2}} \to \mathrm{V_m} \\ (\mathbf{x}, \mathbf{y}) &\to (F \oplus G)(\mathbf{x}, \mathbf{y}) = F(\mathbf{x}) + G(\mathbf{y}).\end{aligned} \tag{20}$$

This is a generalization for vector Boolean functions of the construction of Boolean functions first introduced in Rothaus [1976].

(6) Let $n \geq 1, m = m_1 + m_2, m_1, m_2 \geq 1$ and $F \in \mathcal{F}_{n,m_1}$ and $G \in \mathcal{F}_{n,m_2}$. The result of *adding coordinate functions* of F and G is the function $(F, G) \in \mathcal{F}_{n,m_1+m_2}$, where $(F, G)(\mathbf{x}) = (f_1(\mathbf{x}), \ldots, f_{m_1}(\mathbf{x}), g_1(\mathbf{x}), \ldots, g_{m_2}(\mathbf{x}))$. This is a generalization for vector Boolean functions of the method used in the CAST algorithm and studied in Nyberg [1995] by adding more than one coordinate function at the same time.

(7) Let $n = n_1 + n_2, n_1, n_2 \geq 1, m = m_1 + m_2, m_1, m_2 \geq 1, F \in \mathcal{F}_{n_1,m_1}$, and $G \in \mathcal{F}_{n_2,m_2}$. The *bricklayer* of F and G is the function $F|G \in \mathcal{F}_{n,m}$, where

$$\begin{aligned}(F|G) &: \mathrm{V_{n_1}} \times \mathrm{V_{n_2}} \to \mathrm{V_{m_1}} \times \mathrm{V_{m_2}} \\ (\mathbf{x}, \mathbf{y}) &\to F|G(\mathbf{x}, \mathbf{y}) = (f_1(\mathbf{x}), \ldots, f_{m_1}(\mathbf{x}), g_1(\mathbf{y}), \ldots, g_{m_2}(\mathbf{y})).\end{aligned} \tag{21}$$

This construction corresponds to the bricklayer function [Daemen and Rijmen 2002] as a parallel application of a number of vector Boolean functions operating on smaller inputs. Some authors call this construction concatenation.

Table 5.1 in VBF [2015] lists the member functions related to the previous characterizing elements.

4. EXAMPLES OF APPLICATION OF THE VBF LIBRARY

4.1. KASUMI Cipher Algorithm Evaluation

This section studies the block cipher called KASUMI; it is used in UMTS [2014], GSM [2014], and GPRS [2014] mobile communications systems. UMTS uses KASUMI [2014] in the confidentiality (f8) and integrity (f9) algorithms named UEA1 and UIA1 [KASUMI 2014], respectively. GSM employs KASUMI in the A5/3 key stream generator, whereas GPRS does so in the GEA3 key stream generator.

KASUMI encrypts a 64-bit input by iterating a round function 8 times. The round function consists of the composition of a 32-bit nonlinear mixing function (FO) and a 32-bit linear mixing function (FL). The FO function is again an iterated ladder design consisting of three rounds of a 16-bit nonlinear mixing function FI. In turn, FI is defined as a four-round structure using nonlinear lookup tables S7 and S9. All functions involved will mix the data input with key material.

In the following, each functional component of KASUMI is studied using VBF with the aim to reveal any weakness that could be used as a basis for an attack on the entire algorithm. Such study characterizes the S-boxes as well as the FI function.

4.1.1. S-boxes Characterization. The study of S-boxes $S7$ and $S9$ shows that they are Almost Perfect Nonlinear (APN) bijective Boolean mappings. In fact, a linear approximation analysis shows that the $S7$ nonlinearity (Item $\mathcal{NL}$ in Table II) is equal to 56, which is the maximum value for an S-box with seven input variables. Its linear potential (Item $\mathcal{LP}$ in Table II) is equal to 0.015625 and it has a second-order nonlinearity (Item $\mathcal{NL}_2$ in Table II) of 36. Concerning $S9$, the value for the nonlinearity is equal to 240 over the best-known bound for an S-box with nine input variables, which is 242. Its linear potential is equal to 0.00390625.

From these results, we can conclude that $S7$ offers the best immunity against linear attacks for a 7×7 S-box; in addition, $S9$ immunity against this type of attack is almost optimal.

A differential approximation analysis shows that the $S7$ linearity distance is equal to 28 over a maximum value of 32 and its differential potential is equal to 0.015625. Finally, the value for the linearity distance of $S9$ is equal to 0 and its differential potential is equal to 0.00390625.

From these results, we can conclude that $S7$ and $S9$ do not have an optimal immunity against differential attacks.

The algebraic normal forms of $S7$ and $S9$ are given by Equations (22) and (23), respectively:

$$\begin{aligned}
f_1 &= x_5x_6 + x_4x_6x_7 + x_3x_7 + x_2x_6 + x_2x_4 + x_1 + x_1x_6x_7 + x_1x_4x_5 + x_1x_3x_6 + x_1x_2x_7 \\
f_2 &= 1 + x_5 + x_5x_7 + x_4x_7 + x_4x_5x_6 + x_3x_5x_7 + x_2x_7 + x_2x_5 + x_2x_3 + x_1x_6 + x_1x_5x_6 \\
&\quad + x_1x_4x_7 + x_1x_3x_4 + x_1x_2x_5 \\
f_3 &= 1 + x_5x_7 + x_4 + x_4x_6 + x_3x_6 + x_3x_6x_7 + x_3x_4x_5 + x_2x_7 + x_2x_4x_6 + x_2x_3x_7 + x_1x_6 \\
&\quad + x_1x_4 + x_1x_4x_7 + x_1x_2 \\
f_4 &= x_6 + x_5x_6x_7 + x_3x_6 + x_3x_4 + x_2x_7 + x_2x_6x_7 + x_2x_4x_5 + x_2x_3x_6 + x_1x_5 + x_1x_4x_6 \\
f_5 &= 1 + x_7 + x_4x_7 + x_4x_5 + x_3x_5x_6 + x_3x_4x_7 + x_2x_6 + x_2x_5x_7 + x_1x_7 + x_1x_6x_7 + x_1x_5 \\
&\quad + x_1x_3 \\
f_6 &= 1 + x_6x_7 + x_3x_7 + x_3x_5 + x_2 + x_2x_5x_6 + x_2x_4x_7 + x_1 + x_1x_5x_7 + x_1x_4 + x_1x_2x_3 \\
f_7 &= x_4x_6 + x_3 + x_3x_6x_7 + x_2 + x_2x_5 + x_2x_3x_4 + x_1 + x_1x_7 + x_1x_6 + x_1x_4 + x_1x_3x_5 \\
&\quad + x_1x_2x_6 + x_1x_2x_3
\end{aligned} \tag{22}$$

$$\begin{aligned}
f_1 &= x_8x_9 + x_7 + x_7x_8 + x_5x_6 + x_4x_8 + x_4x_7 + x_3x_8 + x_3x_5 + x_2 + x_1x_7 + x_1x_6 \\
f_2 &= 1 + x_8x_9 + x_7x_9 + x_7x_8 + x_6 + x_6x_9 + x_6x_7 + x_4x_5 + x_3x_7 + x_3x_6 + x_2x_7 + x_2x_4 + x_1 \\
f_3 &= x_9 + x_6x_7 + x_4x_8 + x_4x_7 + x_4x_5 + x_3x_6 + x_3x_5 + x_3x_4 + x_2 + x_1x_8 + x_1x_6 + x_1x_4 \\
&\quad + x_1x_2 \\
f_4 &= 1 + x_7 + x_5x_8 + x_4x_5 + x_3x_9 + x_3x_8 + x_2x_6 + x_2x_5 + x_2x_3 + x_1x_4 + x_1x_3 + x_1x_2 \\
f_5 &= x_8x_9 + x_6x_8 + x_5 + x_4x_9 + x_3x_6 + x_2x_9 + x_2x_3 + x_1x_8 + x_1x_7 + x_1x_6 \\
f_6 &= x_9 + x_7x_8 + x_6x_9 + x_5x_7 + x_4 + x_3x_9 + x_3x_8 + x_2x_5 + x_1x_9 + x_1x_8 + x_1x_2 \\
f_7 &= 1 + x_8 + x_6x_9 + x_5x_6 + x_4x_9 + x_3x_7 + x_3x_6 + x_3x_4 + x_2x_5 + x_2x_4 + x_2x_3 + x_1 + x_1x_9 \\
f_8 &= 1 + x_8 + x_8x_9 + x_6x_7 + x_5x_9 + x_5x_8 + x_4x_9 + x_4x_6 + x_3 + x_2x_8 + x_2x_7 + x_1x_4 \\
f_9 &= 1 + x_7x_9 + x_6 + x_4x_7 + x_3x_4 + x_2x_9 + x_2x_8 + x_2x_7 + x_1x_5 + x_1x_4 + x_1x_2.
\end{aligned} \tag{23}$$

These forms show that the algebraic degree of $S7$ is 3 and the algebraic degree of $S9$ is 2. The algebraic degree of both S-boxes is low and higher-order differential attack can be executed against them. The component algebraic immunity of $S7$ is equal to 3 and for $S9$ is 2. As a consequence, algebraic attacks by solving a system of multivariate algebraic equations can be executed against $S7$ and especially against $S9$.

Concerning the cycle structure, $S7$ and $S9$ have no obvious deficiencies, for example, a large number of transpositions. $S7$ has one fixed point, (0, 0, 1, 1, 0, 1, 1) and has no negated fixed points. $S9$ has one fixed point, (0, 1, 0, 0, 1, 0, 1, 1, 1) and one negated fixed point, (1, 0, 0, 0, 1, 1, 0, 0, 0). The cycle structure of the $S7$ and $S9$ permutations is shown in Table I.

Regarding the second moment of the autocorrelation coefficients, for $S7$ the absolute indicator is 16 and the sum-of-squares indicator is 32,768. For $S9$ the absolute indicator is 512 and the sum-of-squares indicator is 524,288.

From these results, we can conclude that $S7$ achieves a fairly good diffusion since its absolute indicator is nearer the lower theoretical bound, 0, than the upper bound, 128, and similarly for the sum-of-squares indicator where the theoretical bounds are 16,384 and 2,097,152. $S9$ does not achieve a good diffusion because its absolute indicator coincides with the upper bound, 512, while its sum-of-squares indicator is quite close to the lower bound of 262,144.

Table I. Cycle Structure for $S7$ and $S9$

Cycle Length for $S7$	Number of Cycles for $S7$	Cycle Length for $S9$	Number of Cycles for $S9$
1	1	1	2
13	1	2	1
22	1	12	1
92	1	26	1
–	–	74	1
–	–	121	1
–	–	275	1

Table II. $S7$ and $S9$ Cryptographic Criteria

S-box	$\mathcal{NL}$	$\mathcal{NL}_2$	$\mathcal{LD}$	$\mathcal{DEG}$	$\mathcal{AI}$	$\mathcal{MAXAC}$	σ	$\mathcal{LP}$	$\mathcal{DP}$
$S7$	56	36	63	3	3	16	32768	0.015625	0.015625
$S9$	240	0	0	2	2	512	524288	0.00390625	0.00390625

A summary of these criteria is given in Table II.

The Walsh Spectra of the $S7$ and $S9$ mappings are three valued (except from the first row and column value): $16, 0$, and -16 for $S7$, and $32, 0$, and -32 for $S9$. The Linear Profiles of the $S7$ and $S9$ mappings are two-valued (except from the first row and column value): 0 and 256 for $S7$, and 0 and $1,024$ for $S9$. The Differential Profile of the $S7$ and $S9$ mappings are two-valued (except from the first row and column value): 0 and $32,768$ for $S7$, and 0 and $524,288$ for $S9$. The Autocorrelation Spectrum of the $S7$ mapping is four-valued: $128, 16, -16$, and 0 for $S7$, and three-valued for $S9$: $512, -512$, and 0. Having a few-valued WS indicates good cryptographic properties (see Gong et al. [2014]).

The cryptanalysis performed in this section provides, to the best of our knowledge, new results about $S7$ and $S9$ that do not appear in 3rd Generation Partnership Project [2001]. For example, representations of both S-boxes as Truth Table, polynomials in ANF, ANF tables, and Walsh Spectrum are calculated (via the Truth Table); in addition, cryptographic criteria such as nonlinearity, second-order nonlinearity, linearity distance, algebraic immunity, absolute indicator, and sum-of-squares indicator are also computed.

4.1.2. FI Function Characterization. The FI function is a 16×16-vector Boolean function that constitutes the basic randomizing function of KASUMI. It is composed of a four-round structure using the S-boxes $S7$ and $S9$ as shown in Figure 1.

The function FI takes a 16-bit data input I and 16-bit subkey $KI_{i,j}$. The input I is split into two unequal components, a 9-bit left half L_0 and a 7-bit right half R_0, where $I = L_0||R_0$. Similarly, the key $KI_{i,j}$ is split into a 7-bit component $KI_{i,j,1}$ and a 9-bit component $KI_{i,j,2}$, where $KI_{i,j} = KI_{i,j,1}||KI_{i,j,2}$. The function uses two S-boxes, $S7$, which maps a 7-bit input to a 7-bit output, and $S9$, which maps a 9-bit input to a 9-bit output. It also uses two additional functions that are designated $ZE()$ and $TR()$, where $ZE(x)$ takes the 7-bit value x and converts it to a 9-bit value by adding two 0 bits to the most-significant end and $TR(x)$ takes the 9-bit value x and converts it to a 7-bit value by discarding the two most significant bits. The following equations summarize the implementation of function FI:

$$\begin{aligned} I &= L_0||R_0 \\ KI_{i,j} &= KI_{i,j,1}||KI_{i,j,2} \end{aligned} \tag{24}$$

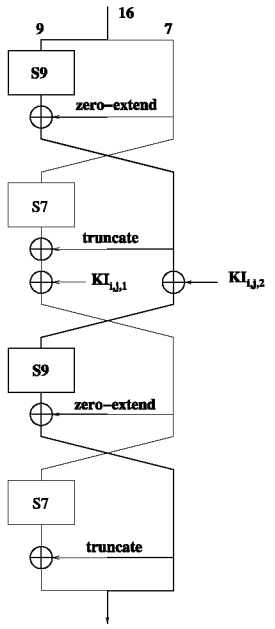

Fig. 1. FI function of KASUMI.

$$\begin{aligned} L_1 &= R_0 \quad R_1 = S9(L_0) + ZE(R_0) \\ L_2 &= R_1 + KI_{i,j,2} \quad R_2 = S7(L_1) + TR(R_1) + KI_{i,j,1} \\ L_3 &= R_2 \quad R_3 = S9(L_2) + ZE(R_2) \\ L_4 &= S7(L_3) + TR(R_3) \quad R_4 = R_3 \end{aligned} \tag{25}$$

$$\begin{aligned} L_4 &= S7(S7(R_0) + TR(S9(L_0) + ZE(R_0)) + KI_{i,j,1}) + TR(S9(S9(L_0) + ZE(R_0) \\ &\quad + KI_{i,j,2}) + ZE(S7(R_0) + TR(S9(L_0) + ZE(R_0)) + KI_{i,j,1})) \\ R_4 &= S9(R_1 + KI_{i,j,2}) + ZE(S7(R_0) + TR(S9(L_0) + ZE(R_0)) + KI_{i,j,1}). \end{aligned} \tag{26}$$

The algebraic degree of the FI function for all the possible 65, 536 values of the key was analyzed. This study reveals that two values of algebraic degree are obtained: 15 and 16 with a frequency of 32, 931 and 32, 605, respectively. Such degrees reveal that FI has a very good resistance against higher-order differential attacks as the maximum possible algebraic degree is 16.

Concerning the cycle structure, the FI function was analyzed for all the possible 65, 536 values of the key. There are key values for which the number of cycles is quite high; for example, the key 0x$a77b$ has the maximum number of cycles, 2, 907. This number of cycles is more than three times the proportion that was present in $S9$. In this case, a higher number of transpositions was expected and it could reveal some kind of deficiency.

For several keys, this function has a significant amount of fixed points and/or negated fixed points. The maximum number of fixed points is six for key values:

0x$57bc$, 0x$5c38$, 0x$6bfe$, 0x$7b4b$, 0x$85c2$, 0x$987e$, 0x$9a32$, 0x$a3ef$, 0x$a5ab$, 0x$acbb$, 0x$b0b4$, 0x$b0e5$, 0x$b327$, 0x$b5c7$, 0x$b90d$, 0x$c4ee$, 0x$c7e4$, 0x$ca74$, 0x$cb5d$, 0x$cb5d$, 0x$cd11$, 0x$cdbe$, 0x$ce24$, 0x$d5da$, 0x$e3ce$, 0x$e4eb$, 0x$e531$, 0x$ea3b$, 0x$ef5c$, 0x$f276$, 0x$f59e$, 0x$fd44$. (27)

Table III. NibbleSub Truth Table

Input	0000	0001	0010	0011	0100	0101	0110	0111	1000	1001	1010	1011	1100	1101	1110	1111
Output	1110	0100	1101	0001	0010	1111	1011	1000	0011	1010	0110	1100	0101	1001	0000	0111

The maximum number of negated fixed points is seven for key values:

$$0x2c3c,\ 0x4041,\ 0x4343,\ 0x06e9,\ 0x518f,\ 0x59ac,\ 0xa161,\ 0xa244,\ 0xab77,\ 0xe1d1,\ 0x1aee,\ 0x1df0. \tag{28}$$

The key value 0xb0b4 has 10 fixed or negated fixed points and a number of key values have nine fixed or negated fixed points:

$$0x3bd0,\ 0x4343,\ 0x5e94,\ 0x5ff3,\ 0x6271,\ 0x682d,\ 0x6e45,\ 0x99e3,\ 0xab77,\ 0xb750,\ 0xc5da,\ 0xd5da,\ 0x167d,\ 0x1df0. \tag{29}$$

This number of fixed and/or negated fixed points is not very high compared with the total number of possible inputs/outputs, 65, 536, and we may, therefore, conclude that the FI function is reasonably well designed.

The Walsh Spectra of FI for several keys have also been computed and from this, the nonlinearities and linear potentials of FI for the 65, 536 keys have been obtained. Nonlinearities range from 31, 534 (with the keys 081e, 2d71, and 52c4) to 32, 049 (with the key c6a6) and linear potentials from 0.00048146 to 0.00141818. Although these nonlinearity values are far from the maximum possible, 32, 640, the linear potentials do not reveal an obvious vulnerability to linear attacks.

As a summary, the KASUMI analysis with VBF provides values for the characteristics analyzed that are similar to those obtained in previous studies [3rd Generation Partnership Project 2001]. In addition, new characteristics such as the algebraic degree, cycle structure, fixed points, negated fixed points, and nonlinearities are also provided. A detailed description of KASUMI cipher analysis within "KASUMI Analysis" in the "Examples" menu can be found in VBFlib [2015].

4.2. Mini-AES Cipher Algorithm Evaluation

Raphael Chung-Wei Phan presented a version of the AES [Phan 2002], with all the parameters significantly reduced while preserving its original structure. This Mini version is purely educational and is designed to grasp the underlying concepts of Rijndael-like ciphers. It may also serve as a testbed for starting cryptanalysts to experiment with various cryptanalytic attacks. The Mini-AES cipher is a 16×16-vector Boolean function and the Mini-AES encryption is performed with a secret key of 16 bits.

4.2.1. S-box Characterization. The Mini-AES S-box is called NibbleSub, and it defines a simple operation that substitutes each input with an output according to a 4×4 substitution table (S-box) given in Table III. These values are, in fact, taken from the first row of the first S-box in DES.

The study of the S-box shows that it defines APN bijective Boolean mappings. In fact, a linear approximation analysis shows that the nonlinearity is equal to 2 while the maximum value for an S-box with four input variables is 6. Its linear potential is equal to 0.5625 and it has a second-order nonlinearity of 0. From these results, we can conclude that $NibbleSub$ does not offer good immunity against linear attacks for a 4×4 S-box.

A differential approximation analysis shows that the $NibbleSub$ linearity distance is equal to 0 over a maximum value of 4 and its differential potential is equal to 0.5. From the previous results, we can conclude that $NibbleSub$ does not have optimal immunity against differential attacks.

Table IV. Cycle Structure

Cycle Length	Number of Cycles
2	1
14	1

Table V. *NibbleSub* Cryptographic Criteria

S-box	$\mathcal{NL}$	$\mathcal{NL}_2$	$\mathcal{LD}$	$\mathcal{DEG}$	$\mathcal{AI}$	$\mathcal{MAXAC}$	σ	$\mathcal{LP}$	$\mathcal{DP}$
NibbleSub	2	0	0	2	2	16	1408	0.5625	0.5

The algebraic normal form of *NibbleSub* is

$$\begin{aligned} f_1 &= 1 + x_4 + x_2 + x_2x_3 + x_2x_3x_4 + x_1 + x_1x_2 + x_1x_2x_3 \\ f_2 &= 1 + x_3x_4 + x_2 + x_2x_4 + x_1 + x_1x_3 + x_1x_3x_4 \\ f_3 &= 1 + x_4 + x_3 + x_3x_4 + x_2x_4 + x_2x_3 + x_1x_4 + x_1x_3 + x_1x_2 + x_1x_2x_4 + x_1x_2x_3 \\ f_4 &= x_3 + x_2x_4 + x_1 + x_1x_4 + x_1x_3x_4. \end{aligned} \tag{30}$$

These forms show that the algebraic degree of *NibbleSub* is 2, which is not high enough to be immune against higher-order differential attacks. The component algebraic immunity is equal to 2: as a consequence, algebraic attacks by solving a system of multivariate algebraic equations can be easily executed.

Concerning the cycle structure, it has no obvious deficiencies, for example, a large number of transpositions; in addition, it has no fixed points and two negated fixed points (0, 0, 1, 0) and (0, 1, 1, 1). The cycle structure is given in Table IV.

Regarding the second moment of the autocorrelation coefficients, the absolute indicator is equal to 16 and the sum-of-squares indicator is 1,408. Hence, *NibbleSub* does not achieve a good diffusion because its absolute indicator reaches the upper bound of 16 while its sum-of-squares indicator is quite close to the upper bound of 4, 096.

A summary of the results for these criteria is represented in Table V.

Excluding the value of the first row and column, the Walsh Spectrum of the *NibbleSub* mapping takes values among 12, 8, 4, 0, −4, −8, and −12; the Linear Profile takes values among 144, 64, 16, and 0; the Differential Profile takes values among 2, 048, 1, 536, 1, 024, 512, and 0; finally, the Autocorrelation Spectrum is five-valued: 16, 8, 0, −8, and –16.

4.2.2. Mini-AES Cipher Characterization. The algebraic degree of Mini-AES for all the possible 65, 536 values of the key was analyzed. This study reveals that only one value of algebraic degree is obtained: 14, which indicates that Mini-AES has a fairly good resistance against higher-order differential attacks since the maximum possible algebraic degree is 16.

In addition, the cycle structure of Mini-AES was analyzed for all the possible 65, 536 values of the key. It was found that no key values provide a high number of cycles: the key (expressed in hexadecimal representation) $0x9e06$ has the maximum number of cycles, 28. No deficiency is expected with respect to this criterion.

For several keys, this cipher has a relevant amount of fixed points and/or negated fixed points. The maximum number of fixed points is 7 for key values 0x0352, 0x4661, 0x5557, and 0x783f. The maximum number of negated fixed points is 9 for key values 0x1$d9b$ and 0x7734. The key values 0x1$d9b$ and 0x4661 have 11 fixed or negated fixed and several values have 10 fixed or negated fixed points:

$$\begin{aligned} &0\text{x}010a,\ 0\text{x}0164,\ 0\text{x}1ce9,\ 0\text{x}24cd,\ 0\text{x}2e38,\ 0\text{x}4a15,\ 0\text{x}5015,\ 0\text{x}7734,\ 0\text{x}783f,\ 0\text{x}9868, \\ &0\text{x}9f18,\ 0\text{x}a8af,\ 0\text{x}aec4,\ 0\text{x}bc85,\ 0\text{x}c9ca,\ 0\text{x}db09,\ 0\text{x}f580. \end{aligned} \tag{31}$$

Table VI. Frequency Distribution of the Absolute Values of the Walsh Spectrum

f	Values
f_1	$(4, 30), (12, 46), (20, 226), (28, 210)$
f_2	$(4, 30), (12, 46), (20, 226), (28, 210)$
f_3	$(4, 30), (12, 46), (20, 226), (28, 210)$
f_4	$(4, 56), (12, 58), (20, 154), (28, 244)$
f_5	$(4, 57), (12, 91), (20, 97), (28, 267)$

Table VII. Frequency Distribution of the Absolute Values of the Autocorrelation Spectrum

f	Values
f_1	$(0, 129), (8, 298), (16, 60), (24, 9), (32, 2), (40, 13), (512, 1)$
f_2	$(0, 150), (8, 196), (16, 148), (24, 12), (32, 5), (512, 1)$
f_3	$(0, 183), (8, 223), (16, 84), (24, 6), (32, 4), (40, 10), (56, 1), (512, 1)$
f_4	$(0, 157), (8, 232), (16, 84), (24, 8), (32, 17), (40, 10), (48, 3), (512, 1)$
f_5	$(0, 192), (8, 156), (16, 129), (24, 9), (32, 13), (40, 3), (48, 6), (64, 3), (512, 1)$

This number of fixed and/or negated fixed points is not very high when compared with the total number of possible inputs/outputs (65, 536). We can conclude that the Mini-AES cipher is reasonably well designed from this point of view.

Several Walsh Spectra of Mini-AES for different keys have also been computed, and from these the nonlinearities and linear potentials of Mini-AES for more of the 65, 536 keys have been obtained. Nonlinearities range from 31, 432 (with the key $69b0$) to 32, 040 (with the key $f7de$) and linear potentials from 0.000493586 to 0.001662314. Although these nonlinearities are far from the maximum possible nonlinearity, 32, 640, the linear potentials do not reveal an obvious vulnerability to linear attacks.

4.3. Search for Vector Boolean Functions with Excellent Profiles

Boolean functions with very high nonlinearity pose some of the most challenging problems in the area of symmetric cryptography and combinatorics. For functions with an even number of variables, n, the maximum possible nonlinearity $2^{n-1} - 2^{n/2-1}$ is attained for the well-known bent functions. However, for the case when n is odd, constructing Boolean functions with maximum possible nonlinearity is an unsettled open problem. Hence, so far suboptimal results have been mainly obtained via heuristic search; for example, for $n = 9$ the best-known nonlinearity result is 242 [Kavut and Yucel 2010].

In this section, we illustrate how the VBF library can be employed to apply a steepest descent for a search of nine-variable Boolean functions with the highest nonlinearity. Using this algorithm, we have found 5, 121 Boolean functions with nonlinearity 242, which can be grouped into five different affine equivalence classes. Two Boolean functions f, g are affine equivalent if the following equality holds:

$$g(\mathbf{x}) = f(A\mathbf{x} + \mathbf{b}) + \mathbf{c}\mathbf{x} + d, \tag{32}$$

where $A \in \mathsf{M}_{n\times n}(\mathrm{GF}(2))$, $\mathbf{b}, \mathbf{c} \in \mathrm{V_n}$, and $d \in \mathrm{GF}(2)$.

There are some function properties that are invariant over the mapping defined in Equation (32). In fact, the five obtained affine equivalence classes can be identified by invariant properties such as the frequency distribution of the absolute values of the Walsh Spectrum or the Autocorrelation Spectrum. These invariants have been obtained for the five affine equivalence classes with the VBF library, providing the results shown in Table VI.

We have found millions Boolean functions within the classes f_1, f_2, f_3, f_4, f_5. The Truth Tables of all these Boolean functions are available from VBFlib [2015].

Table VIII. Additional Cryptographic Criteria

Class	$\mathcal{LD}$	$\mathcal{DEG}$	$\mathcal{AI}$	$\mathcal{MAXAC}$	σ
f_1	118	7	4	40	324608
f_2	120–122	7	4	32	324608
f_3	114, 118	7	4	56	324608
f_4	116, 118	7	4	48	343424
f_5	112	7	4	64	354560

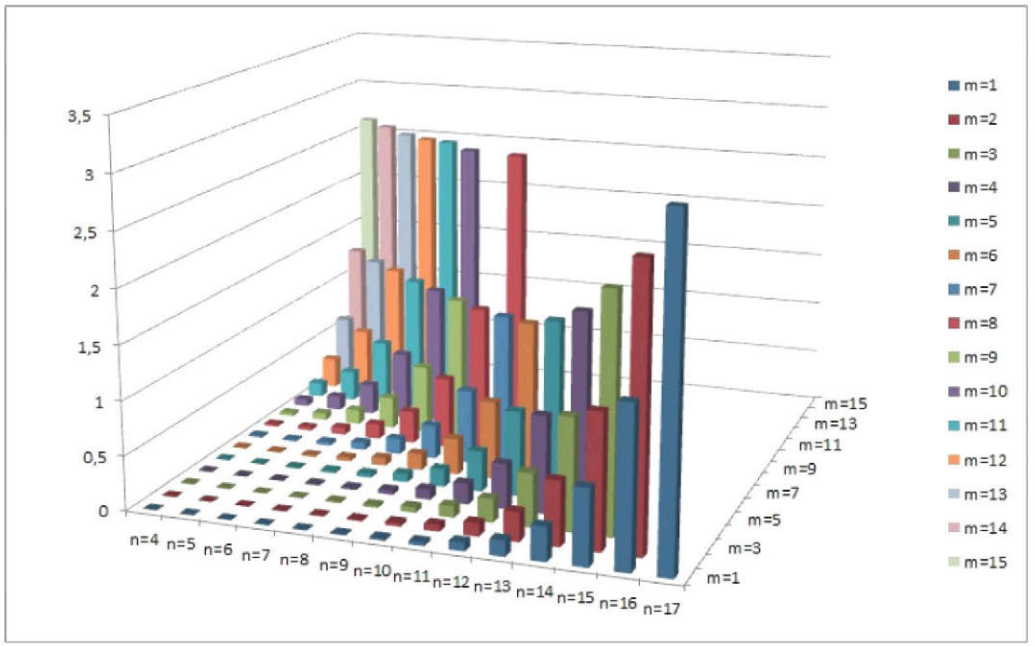

Fig. 2. Overall CPU time in seconds for cryptographic characterization of $n \times m$ S-boxes.

Using the VBF library, the value of other cryptographic criteria (linearity distance, second-order nonlinearity, algebraic degree, algebraic immunity, absolute indicator, and sum-of-squares indicator) are easily computed for each one of these Boolean functions. Table VIII shows the range of values such criteria take for the functions within each class.

This information allows for an appropriate function selection for cipher design. For instance, among all these functions with the highest nonlinearity, we can determine those with highest linearity distance ($\mathcal{LD} = 122$, located in the second class). The hexadecimal representation of the Truth Tables of these finally selected two functions are

$$B8FE8F795F6CDA63FA26AC2B2EBB477B7058C266BE53DC0480DF6BFCB8A70E54A4E7EFDF91788517C9CF410DA90A10EBCE7A663C2B1F4B2C634DA1C1DE5C54AA$$

$$E2A42ADCFAC980395F83F67174E1E2DED5FD983CE40979A1DA85CE591D02540EFEBD4A7A34DDDF4D6C6A1B57F350B54E6BDF3C667145EE89391704647BF90EF0.$$

These specific functions may be used in the construction of high-performance S-boxes for cipher design.

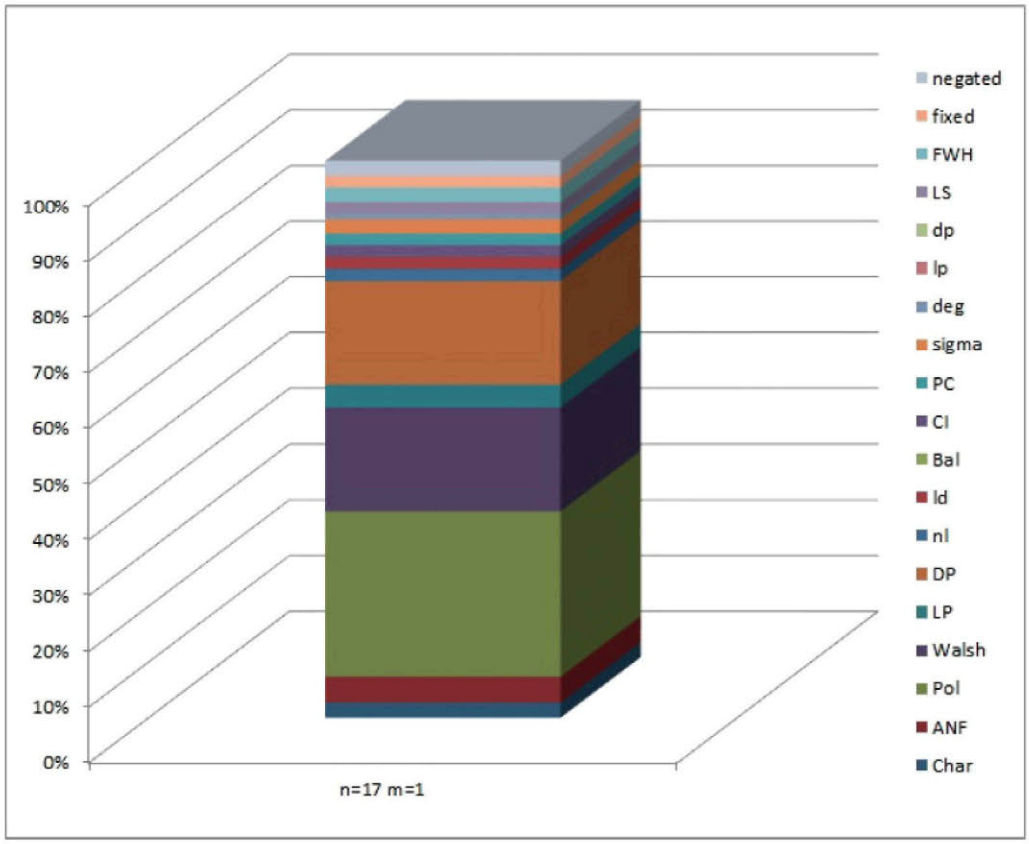

Fig. 3. CPU timing measurements for all functions in Algorithm 1.

5. COMPUTATIONAL COST RESULTS

This section assesses and displays the computational performance of the VBF library algorithms via its application to a cryptographic characterization of S-boxes with different size $n \times m$. The program, described in Algorithm I in VBF [2015], first generates random Truth Tables, which correspond to vector Boolean functions with dimensions ranging from n = 4 to n = 17 and from m = 1 to m = 15. It then calculates the cryptographic criteria of the vector Boolean functions given from these Truth Tables. The calculations were performed on an Intel(R) Core(TM) i7-2600K CPU @3.40GHz, 16GB RAM, 1TB Debian Linux. Figure 2 presents the corresponding computing times graphically, which happen to be a good measure of the computational complexity, provided the computer resources are not exhausted. (If $m + n$ is too large for the whole computation to take place in the available RAM, then the compute time will increase drastically due to hard disk swapping.)

As expected, in normal computer conditions, the computational complexity grows exponentially in the bit length of the vector Boolean functions.

In Figure 3, we present detailed timing measurements corresponding to the different functions tested for the values $n = 17$ and $m = 1$. This heterogeneous distribution of the computing times reflects the diverse complexity of the different routines provided by the VBF library.

6. CONCLUSIONS

In this article, a C++ library designed to analyze vector Boolean functions from a cryptographic perspective has been presented. It represents a very useful tool for analyzing cryptographic primitives expressed as vector Boolean functions in a very efficient way. This class supports as input a broad range of vector Boolean function representation data structures such as Truth Tables, ANF tables, polynomials in ANF, trace representations, permutation and linear matrices, and DES-like S-boxes. Then it can provide new structures that represent cryptographic criteria such as the Walsh Spectrum, Differential Profile, and Autocorrelation Spectrum, among others. Cryptographic criteria such as nonlinearity, linearity distance, correlation immunity, balancedness, algebraic degree, algebraic immunity, and propagation criterion are easily obtained. The behavior of the cryptographic properties of secondary constructions resulting from operations over vector Boolean functions can also be studied by means of the VBF class. The applicability of the library has been illustrated by analyzing real systems such as the KASUMI and Mini-AES block ciphers, and by determining Boolean functions with very high nonlinearity that are robust against linear attacks.

REFERENCES

3rd Generation Partnership Project. 2001. *Security Algorithms Group of Experts (SAGE); Report on the Evaluation of 3GPP Standard Confidentiality and Integrity Algorithms (SAGE version 2.0)*. Technical Report. 3GPP.

3rd Generation Partnership Project. 2005. *Specification of the 3GPP Confidentiality and Integrity Algorithms - Document 2: KASUMI Specification (Release 6) No. 3GPP TS 35.202 V6.1.0 (2005-09)*. Technical Report. 3GPP.

C. M. Adams and S. E. Tavares. 1993. Designing s-boxes for ciphers resistant to differential cryptanalysis (extended abstract). In *Proceedings of the 3rd Symposium on State and Progress of Research in Cryptography*. 181–190.

J. A. Álvarez-Cubero and P. J. Zufiria. 2010. A C++ class for analysing vector boolean functions from a cryptographic perspective. In *Proceedings of the International Conference on Security and Cryptography (SECRYPT '10), SECRYPT is part of The International Joint Conference on e-Business and Telecommunications*, Sokratis K. Katsikas and Pierangela Samarati (Eds.). SciTePress, 512–520.

J. A. Álvarez-Cubero and P. J. Zufiria. 2012. Cryptographic criteria on vector boolean functions. In *Cryptography and Security in Computing*, Jaydip Sen (Ed.). InTech, 51–70.

A. Bibliowicz, P. Cohen, and E. Biham. 2003. *A System for Assisting Analysis of some Block Ciphers*. Technical Report NES/DOC/TEC/WP2/007/2. Israel Institute of Technology, Haifa, Israel.

E. Biham and A. Shamir. 1990. Differential cryptanalysis of DES-like cryptosystems. In *Proceedings of the 10th Annual International Cryptology Conference on Advances in Cryptology (CRYPTO'90)*. 2–21. DOI:http://dx.doi.org/10.1007/3-540-38424-3_1

BOOLFUN 2010. Cryptographic Boolean Functions. Retrieved from http://cran.r-project.org/web/packages/boolfun/index.html.

C. Carlet. 2004. On the secondary constructions of resilient and bent functions. *Progress in Computer Science and Applied Logic* 23 (2004), 3–28.

C. Carlet. 2008a. *Boolean Functions for Cryptography and Error Correcting Codes*. Technical Report. University of Paris, BP 105-78153, Le Chesnay Cedex, FRANCE.

C. Carlet. 2008b. On the higher order nonlinearities of boolean functions and s-boxes, and their generalizations. In *Sequences and Their Applications (SETA'08)*. Springer, New York, NY, 345–367.

C. Carlet. 2008c. *Vectorial Boolean functions for Cryptography*. Technical Report. University of Paris, BP 105-78153, Le Chesnay Cedex, FRANCE.

F. Chabaud and S. Vaudenay. 1995. Links between differential and linear cryptanalysis. In *Advances in Cryptology (EUROCRYPT'94) (Lecture Notes in Computer Science)*, Alfredo De Santis (Ed.), Vol. 950. Springer, Berlin, 356–365.

N. Courtois. 2003. Fast algebraic attacks on stream ciphers with linear feedback. In *Advances in Cryptology (CRYPTO'03) (Lecture Notes in Computer Science)*, Vol. 2729. Springer, Berlin, 177–194.

N. Courtois and W. Meier. 2002. Algebraic attacks on stream ciphers with linear feedback. In *Advances in Cryptology (EUROCRYPT'03) (Lecture Notes in Computer Science)*, Vol. 2656. Springer, Berlin, 346–359.

N. T. Courtois and J. Pieprzyk. 2002. Cryptanalysis of block ciphers with overdefined systems of equations. In *Advances in Cryptology (ASIACRYPT'02)*. Lecture Notes in Computer Science, Vol. 2501. Springer, Berlin, 267–287.

CRYPTOOL 2014. Educational Framework for Cryptography and Cryptanalysis. Retrieved from http://www.cryptool.org/.

J. Daemen and V. Rijmen. 2002. *The Design of Rijndael*. Springer-Verlag New York, Secaucus, NJ.

M. Davio, A. Thayse, and J. P. Deschamps. 1978. *Discrete and Switching Functions*. McGraw-Hill, New York, NY.

J.-H. Evertse. 1988. Linear structures in blockciphers. In *Advances in Cryptology (EUROCRYPT'87*, David Chaum and WynL. Price (Eds.). Lecture Notes in Computer Science, Vol. 304. Springer, Berlin, 249–266. DOI:http://dx.doi.org/10.1007/3-540-39118-5_23

J.-C. Faugère and G. Ars. 2003. *An Algebraic Cryptanalysis of Nonlinear Filter Generators Using Gröbner Bases*. Technical Report. INRIA 4739.

G. Gong, T. Helleseth, H. Hu, and C. Li. 2014. New three-valued walsh transforms from decimations of helleseth-gong sequences. In *Sequences and Their Applications, (SETA'12) (Lecture Notes in Computer Science)*, Vol. 7280. Springer, Berlin, 327–337.

GPRS 2014. General Packet Radio Service. Retrievevd from http://www.etsi.org/index.php/technologies-clusters/technologies/mobile/gprs.

GSM 2014. Global System for Mobile Communications. Retrieved from http://www.etsi.org/index.php/technologies-clusters/technologies/mobile/gsm.

T. Jakobsen and L.R. Knudsen. 1997. The interpolation attack on block ciphers. In *Proceedings of Fast Software Encryption (Lecture Notes in Computer Science)*, Vol. 1267. Springer, Berlin, 28–40.

KASUMI 2014. Block Cipher Used in UMTS, GSM and GPRS. Retrieved from http://www.3gpp.org/DynaReport/35202.htm.

S. Kavut and M. D. Yucel. 2010. 9-variable boolean functions with nonlinearity 242 in the generalized rotation symmetric class. *Information and Computation* 208, 4 (April 2010), 341–350.

X. Lai. 1990. Linear structures of functions over prime fields. (1990). Unpublished.

X. Lai. 1994. Higher order derivatives and differential cryptanalysis. In *Communications and Cryptography*, Richard E. Blahut, Jr. Costello, Daniel J., Ueli Maurer, and Thomas Mittelholzer (Eds.). The Springer International Series in Engineering and Computer Science, Vol. 276. Springer, Berlin, 227–233. DOI:http://dx.doi.org/10.1007/978-1-4615-2694-0_23

F. J. MacWilliams and N. J. A. Sloane. 1977. *The Theory of Error Correcting Codes*. Number pts. 1-2 in North-Holland Mathematical Library. North-Holland Publishing Company, Amsterdam, Netherlands.

MatPack 2006. MatPack C++ Numerics and Graphics Library. Retrieved from http://www.matpack.de/.

M. Matsui. 1994. Linear cryptanalysis method for DES cipher. In *Advances in Cryptology (EUROCRYPT'93)*, Tor Helleseth (Ed.). Lecture Notes in Computer Science, Vol. 765. Springer, Berlin, 386–397. DOI:http://dx.doi.org/10.1007/3-540-48285-7_33

W. Meier and O. Staffelbach. 1990. Nonlinearity criteria for cryptographic functions. In *Advances in Cryptology (EUROCRYPT'89)*, Jean-Jacques Quisquater and Joos Vandewalle (Eds.). Lecture Notes in Computer Science, Vol. 434. Springer, Berlin, 549–562. DOI:http://dx.doi.org/10.1007/3-540-46885-4_53

NTL 2015. NTL: A Library for doing Number Theory. Version 9.2.0 (2015.5.23). Retrieved from http://www.shoup.net/ntl/.

K. Nyberg. 1991. Perfect nonlinear s-boxes. In *Advances in Cryptology (EUROCRYPT'91)*, Donald W. Davies (Ed.). Lecture Notes in Computer Science, Vol. 547. Springer, Berlin, 378–386. DOI:http://dx.doi.org/10.1007/3-540-46416-6_32

K. Nyberg. 1993. On the construction of highly nonlinear permutations. In *Advances in Cryptology (EUROCRYPT'92)*, Rainer A. Rueppel (Ed.). Lecture Notes in Computer Science, Vol. 658. Springer, Berlin, 92–98. DOI:http://dx.doi.org/10.1007/3-540-47555-9_8

K. Nyberg. 1995. S-boxes and round functions with controllable linearity and differential uniformity. In *Fast Software Encryption*, Bart Preneel (Ed.). Lecture Notes in Computer Science, Vol. 1008. Springer, Berlin, 111–130.

National Institute of Standards and Technology. 1999. *FIPS PUB 46-3: Data Encryption Standard (DES)*. National Institute for Standards and Technology, Gaithersburg, MD.

R. C.-W. Phan. 2002. Mini advanced encryption standard (mini-AES): A testbed for cryptanalysis. *Cryptologia* 26, 4 (Oct. 2002), 283–306.

K. Pommerening. 2005a. Fourier Analysis and Boolean Maps – A Tutorial. (2005). http://www.staff.unimainz.de/pommeren/Kryptologie/Bitblock/A_Nonlin/Fourier.pdf.

K. Pommerening. 2005b. *Linearitätsmaße für* Boole *sche Abbildungen*. Technical Report. Fachbereich Mathematik der Johannes-Gutenberg-Universität.

O. S. Rothaus. 1976. On "bent" functions. *Journal of Combinatorial Theory, Series A* 20, 3 (1976), 300–305.

SAGE 2014. Open-source mathematics software. Retrieved from http://www.sagemath.org.

T. Siegenthaler. 1985. Decrypting a class of stream ciphers using ciphertext only. *IEEE Transactions on Computers* 34, 1 (1985), 81–85.

UMTS. 2014. Universal Mobile Telecommunications System. Retrieved from http://www.3gpp.org/Dyna Report/25816.htm.

VBF. 2015. Vector Boolean Functions (VBF) Library. User Manual. (2015).

VBFlib. 2015. VBF: Vector Boolean Functions Library: User Manual and Analysis of Cryptanalytic Algorithms. Retrieved from http://vbflibrary.tk.

X.-M. Zhang and Y. Zheng. 1995. GAC—the criterion for global avalanche characteristics of cryptographic functions. *Journal of Universal Computer Science* 1, 5 (1995), 320–337.